

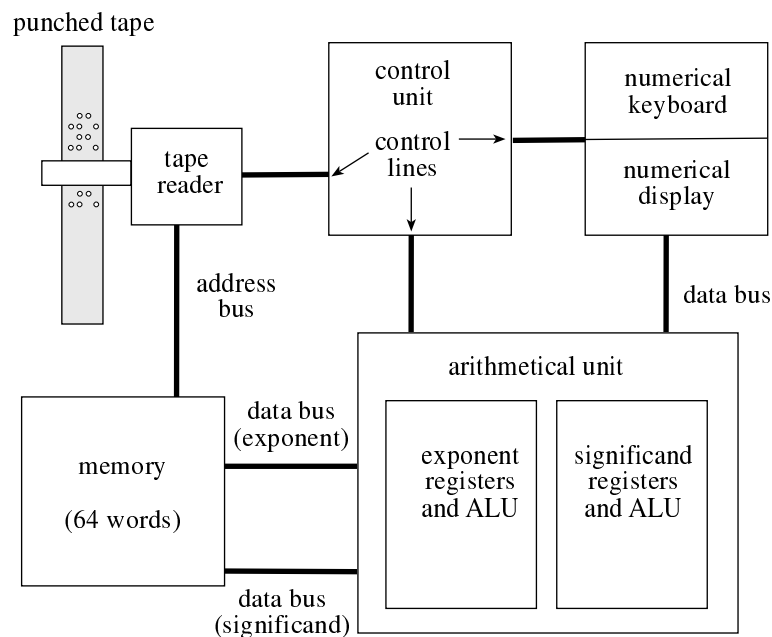
## Z1, Z2, Z3 and Z4

The German inventor Konrad Zuse (1910–1995) built the machines Z1 (1936–1938), Z2 (1940), Z3 (1938–1941), and Z4 (1941–1945) in Berlin, Germany ~ they are among of the earliest computers built in the world. The Z1 was a mechanical device, whereas the Z3, which had the same logical structure, was built using telephone relays. The Z2 was an intermediate prototype in which the memory consisted of mechanical components and the processor was made of relays. The Z4, finally, was also a relay machine and the ^better Z3~, in the sense that it had a much greater memory and the instruction set was enlarged to handle more operations.

All machines used the binary system internally for the arithmetical operations. Numbers were entered in decimal notation and the processor took care of transforming from the decimal to the binary system. After computing the result using binary arithmetic, it was transformed from binary to decimal and was displayed using lamps.

The Z1, Z3 and Z4 worked with floating–point numbers. Zuse wanted his machines to be used for engineering and business computations and developed an internal numerical representation that strongly resembles today’s IEEE floating–point format. Each number was stored using three fields: the sign of the number, the exponent of the number in two’s complement notation, and the mantissa of the number. The number 2.0, for example, was stored as a positive sign, binary exponent 1 and mantissa 1.0, i.e.,  $2 = 1.0 \times 2^1$ . The processor of the Z1, Z3 and Z4 consisted therefore of two main components, one for handling the exponent of a number and one for handling the mantissa.

The three main machines (the Z2 was just an experimental prototype) shared a common architecture. The figure below shows the main components: the memory (64 words for the Z1 and Z3, 1024 for the Z4) was used to store floating–point numbers. The processor operated on them using two internal registers, R1 and R2. The program commands were stored on a punched tape and they were read sequentially. The instructions were executed by the control unit and the I/O console allowed the user to enter decimal numbers through a keyboard or read decimal results from lamps.



Block architecture of the Z1, Z3 and Z4

The clear separation between memory and processor, visible in the figure and not present in machines like the ENIAC or the Harvard Mark I, makes the Z1 and Z3 forerunners of what later was to be called *von Neumann* architecture. However, the program was not stored in memory, mainly because all machines had so few addressable memory locations. Zuse had thought of stored program architectures but did not pursue the idea further.

The instruction set of the Z1 and Z3 consisted of the following operations:

- ◆ addition
- ◆ subtraction
- ◆ multiplication
- ◆ division
- ◆ square root
- ◆ read a number from the keyboard to a register
- ◆ display the number in register R1
- ◆ store a number in memory
- ◆ load a number from memory

Before doing any computation, numbers were read from the numerical keyboard to register R1 and R2. Once the registers were loaded, an addition, subtraction, multiplication or division could be performed. The result was stored again in register R1. The time required by the Z3 for a multiplication was 3 seconds. According to Zuse, the Z1 and Z3 were almost equally fast. The Z3 used 2000 relays, i.e., 600 for the processor and 1400 for the memory.

Using the instruction set shown above, it is possible to process any arithmetical formula of the kind used in engineering applications. However, the instruction set does not provide a conditional branching instruction, so that it is relatively difficult, although not unfeasible, to perform more complex computations. Also, the punched tape can be bound to form a loop, so that iterative calculations are possible. Instructions were punched on the tape using an 8-bit code. Zuse's machines could be driven by the program stored in the tape or could be used interactively, much like modern pocket calculators.

The processor registers R1 and R2 could also be loaded with numbers stored in memory. Adding two numbers stored at the memory locations X and Y, and storing the result in the memory location Z, consisted of the following sequence of operations: a) load R1 with the number stored in address X, b) load R2 with the number stored in address Y, c) perform the addition, d) store the result in address Z. It is not difficult to see how more complex programs could be written for these machines.

Zuse avoided having to use many logical gates for the processor by using a control unit which worked as microsequencers, one for each command in the instruction set. A microsequencer consisted, in the case of the Z3, of a rotating arm which advanced one step in each cycle of the machine. A clock (a rotating motor) provided the clock cycles needed to synchronize the machine. In the case of the Z3, the operating frequency was around 5 Hz. At each cycle, the rotating arm in a microsequencer activated the necessary circuits for the operation at hand. For example, in the case of multiplication, repeated addition and shifting of binary numbers were needed. The 18 partial operations needed were all started by a microsequencer with 18 contacts for the rotating arm. The microsequencer can be thought of as a kind of hardwired program that implemented more complex instructions out of atomic binary operations. Modifying the internal operation of the machine consisted therefore only in rewiring these microsequencers, without having to modify the rest of the processor. This resulted in a very efficient and flexible architecture, as was shown thereafter, when the instruction set of the Z3 was expanded for the Z4.

The Z4 was to become the fully fledged computer that Zuse had had in mind during all those years. It was built under contract for the German Airspace Research Office. The main difference with the Z3 was the larger memory (1024 addresses). However, using relays for the memory was out of the question, since this would have made the Z4 too bulky. Zuse developed a mechanical memory using the same kind of components first used in the Z1. The final size of the larger memory was less than one cubic meter. The processor, on the other hand, was made of relays and although it had the same basic structure as the processor of the Z3, made more instructions available for the programmer. It was possible, for example, to multiply a number by ten, to shift a binary number, to calculate the reciprocal of a number, etc. All these extra instructions did not demand a fundamental change of the architecture of the machine, only to include more microsequencing units in the machine. Multiplication by ten was already effectively present as atomic operation in the Z3 (and was necessary to transform a number from the binary to the decimal notation) but was hidden from the programmer. In the Z4 many hidden instructions were made visible to the programmer, increasing in this way the usefulness of the machine.

But the main difference between the Z4 on the one side, and the Z1 and Z3 on the other, was the inclusion later of conditional branching in the Z4. After the war, the Z4 was moved from Berlin to Bavaria, where it stayed in a barn for almost four years. Prof. Stiefel from the Technical University of Zürich heard of the machine and after visiting Zuse, decided to rent the computer for his university. He asked Zuse to include conditional branching in the instruction set, a second punched tape for numbers and to use a typewriter to print out the results. The machine was refurbished and conditional branching was added. The new instruction worked in the following way: when the contents of register R1 was negative, the control unit skipped all following instructions until a special code ( $\hat{\text{start}}$ ) was found in the punched tape. In this way it was possible to jump over sequences of instructions, i.e., those constituting the not-taken branch of a conditional instruction in the source code.

A detailed comparison of the numerical algorithms used by Zuse for the floating-point calculations in his machines with modern algorithms shows that they are almost identical. For

division, Zuse used the approach now known as non-restoring division. Square-rooting was done using a similar method based on the division algorithm.

During the years 1941–1949 Zuse developed plans for other machines and even applied for patents for some of them. He designed a logarithmic machine, in which numbers were stored as binary logarithms. He also penciled down the circuits for a device which he called a "logic machine", in which each memory word consisted of a single bit. The processor could only perform atomic logic operations (conjunction, disjunction and negation) and a program was a sequence of such atomic operations. Perhaps the most surprising device he thought of was a machine which would be a front-end for the Z4: formulas would be typed in a keyboard, using the standard mathematical notation, and the machine would transform them into assembler code for the Z4. This "Planfertigungsgerät", which was never completed, would have been a hardware compiler.

### References

Zuse, Konrad. *The Computer ~ My Life*. Berlin: Springer-Verlag, 1993.

Rojas, Raul, and Ulf Hashagen (eds.). *The First Computers ~ History and Architectures*. Cambridge, Mass.: MIT Press, 1999.

–RAUL ROJAS